APPROACHES FOR WORKFORCE SCHEDULING PROBLEMS

A Thesis Submitted
In partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
RAVINDRA K PATIL

to the

Inter-disciplinary Programme in Industrial and Management Engineering
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
MARCH, 1976

CERTIFICATE

Certified that the present work on 'Approaches for Workforce Scheduling Problems' by Ravindra Krishnarao Patil has been carried out under my supervision and has not been submitted elsewhere for the award of a degree.

(Jawahar L. Batra)
Convener
Industrial and Management Engg.,
Indian Institute of Technology,
Kanpur 208016

ACKNOWLEDGEMENTS

The author feels elated in manifesting his profound sense of gratitude to Dr. Jawahar L Batra for his kind, able and dynamic guidance in the inception, execution and completion of this thesis. His lively interest, constructive criticism and all round direction has been a perennial source of inspiration and information to me. His constant encouragement and amiable counsel were the sustaining factors throughout the course of this work.

R. K. Patil

jkm/RKP

TABLE OF CONTENTS

LIST OF TABLES

SYNOPSIS

Day-off scheduling is an important area of manpower planning which has attracted the attention of researchers in the field of management sciences. The present work is another contribution to this field.

Certain types of facilities operate seven days each week and face the problem of allocating manpower during the week as staffing requirements fluctuate. This thesis deals with the following particular situation. Employees work five days a week, and operating week is of seven days A day-off schedule for the five-day workweek employees is to be developed, satisfying daily varying manpower requirements. Two days off are to be given per week. They can either be given consecutively or nonconsecutively.

Three different approaches have been proposed in this work. The approaches are. Requirements Reduction Technique (RRT), Artificially Increased Requirements Technique (AIRT), and Status Matrix Technique (SMT).

In the Requirements Reduction Technique, an optimal schedule which allows two consecutive days off and covers all manpower requirements is developed. The problem is structured as a linear programme. The special mathematical structure of the linear programme is exploited to generate an optimal solution. The Artificially Increased Requirements Technique, generates alternate optimal solutions which are used for developing rotating schedules. The rotating schedules provide equitable consecutive day-off assignment to every worker. The Status Matrix Technique, utilizes the possibility of non-consecutive day-off assignment and removes any inherent slack, which might be present, due to consecutive day-off constraint. -Special features like employee preferences, rankings, overtime, etc. can be included in this approach

The methodologies, RRT and AIRT, have been tested on the problems available in the literature. It has been found that RRT requires lesser computational effort than Tibrewala, Philippe and Browne's algorithm. The main feature of AIRT is its capability to produce rotating schedules without restoring to computer facility. This is because the proposed algorithm is simpler as compared to algorithms available in literature. Moreover, AIRT yields optimal schedules. There is no algorithm available in literature which permits the generation of non-consecutive day-off schedules considering the special features like employee preferences, rankings, overtime, etc. Therefore, the computational effectiveness of SMT could not be compared. The proposed algorithms are very simple computationally and involve only hand-calculations.

CHAPTER I

SYSTEM OVERVIEW

Productivity of an organization can be increased by the efficient utilization of its available resources. One of the most important resources for the production of goods and services is the human resource. The determination of ways and means for the efficient utilization of human resource, i.e., workforce has been an area of major interest to production and operations managers. Operations research, organization and methods study, financial analysis, market research and other staff functions are all ultimately concerned with making more effective use of human and capital resources.

It is gradually becoming accepted that manpower is a resource like capital and material, but the effective deployment of human resources is as difficult to master as, the effective deployment of physical and financial resources. All can give valuable returns in terms of a company's or nation's well being; but all can be squandered on trivial activities. Yet the human resource has been largely neglected: most companies think hard before investing ten lacs rupees in equipment, but give scant consideration to the commitment to a similar sum in the 30 years career of one of their executives. Yet by ensuring that employees are at all times contributing to the firm's future viability, not only is the best use made of the resource by the firm but the individuals can receive the maximum amount of job satisfaction for themselves.

The scientific studies pertaining to the intelligent allocation and utilization of human resources are called 'Manpower Planning' studies. One can say that manpower planning is a process by which a firm ensures that it has the right number of people, and the right kind of people, in the right places, at the right time, doing things for which they are economically most useful. Most manpower planning studies are based on the assumption that if the relevant future is sensibly assessed in a disciplined way, then there is a better chance of efficient utilization of human resources. The basic input in manpower planning is the demand forecast. Based on demand forecast, the number of man-days; required number of shifts, and daily workforce requirements are determined.

There are several instances when the facility must operate seven days a week but due to government legislation or trade union requirements, the regular workforce cannot be asked to work seven days a week without any off day in between. Typical real-world examples of seven day operating week include scheduling of workforce in public services (police petrol, ater works, etc.), transportation organizations (baggage andlers, bus drivers, etc.), and sometimes in manufacturing irms, especially those utilizing high capital cost equipment chemical, steel, etc.). Plant maintenance personnel must work throughout the week to keep the machines in working condition. It is also possible that manpower requirements fluctuate during the week.

The most important problem in case of scheduling manpower to meet varying requirements during the week, is the assignment of day-off. Employees may be required to work five or six days a week. In India, most of the organizations observe six-day working week. Due to various reasons, many industrial organizations in the country have already switched over to five-day workweek, despite the fact that seven-day operating week is observed.

The scheduling of six-day workweek employees to seven-day operating week is a very simple problem. However, when five-day workweek is observed in an organization having seven-day operating week, certain important features have to be considered for the development of workers' schedule. Employees are to be given two days off. They can either be given off on two consecutive or nonconsecutive days. Sometimes, for the sake of uniformity employees prefer that the days off be given on rotational as well as consecutive basis.

Although wide spectrum of literature is available on manpower planning, there seems to be a clear deficiency in the literature in the scheduling of five-day week workforce on the seven-day operating week basis. The objective of this thesis is to propose methodologies for the development of schedules which would minimize the total number of 'five-day' workers, while satisfying the daily varying workforce requirements.

For the stated manpower scheduling problems, three approaches have been proposed. These approaches have been titled as,

1 Requirements Reduction Technique (RRT),

2 Artificially Increased Requirements Technique (AIRT),

3 Status Matrix Technique (SMT).

In the Requirements Reduction Technique, an optimal schedule which allows two consecutive days off and covers all manpower requirements is developed. The problem is structured as a linear programme. The special mathematical structure of the linear programme is exploited to generate an optimal solution. The Artificially Increased Requirements Technique, generates alternate optimal solutions which are used for developing rotating schedules. The rotating schedules provide equitable consecutive day-off assignment to every worker. The Status Matrix Technique, utilizes the possibility of nonconsecutive day-off assignment and removes any inherent slack, which might be present, due to consecutive day-off constraint. Special features like employee preferences, rankings, overtime, etc. are included in this approach.

In Chapter II, the state of art report on the literature available in the areas of day-off scheduling and rotating schedules is presented. In Chapter III, the mathematical formulations of the proposed three approaches is given. Illustrative problems are added to explain the various decision making steps involved in the developed algorithms. The conclusions and future scope for research are outlined in Chapter IV.

CHAPTER II

STATE OF ART

The literature on employee day-off assignment can be broadly grouped into the following two categories. 1) Day-off scheduling, and 2) Rotating Schedules.

2.1 Day-off Scheduling:

The literature on day-off scheduling can be divided into: 1) Scheduling with nonconsecutive days off, and 2) Scheduling with consecutive days off.

The literature on scheduling with nonconsecutive days off is discussed first.

Monroe [13] provides a technique for developing weekly personnel assignments with respect to a fixed working shift He first determines the number of persons required on each of the seven days of the week. The total number of mandays per week is rounded to the next multiple of 5 (if necessary) and divided by 5, to obtain an integer, say W, representing the number of workers required.

It is assumed that each worker must have two days off during the week, not necessarily consecutive. The number of workers having Regular Days Off (RDO's) on each day is the difference between the workforce size, and the number of men required on that day. Monroe's technique determines W pairs of RDO's, one pair for each member of the workforce, such that the required number of RDO's are met exactly. Some of the RDO pairs, thus determined, are consecutive and some are composed of nonconsecutive days. Monroe's objective is to produce a solution which maximizes the number of consecutive pairs. The technique entails as much intuition as analysis because of its heuristic nature

A linear programming approach to this problem has been proposed by Rothstein [14]. In his formulation he considered 15 linear constraints involving 15 variables. The first constraint states that the number of workers having Monday-Tuesday off, plus the number of workers having Tuesday-Wednesday off, plus the number of workers assigned nonconsecutively Tuesday off, must equal to the total number of workers assigned Tuesday off. Constraints 2 to 7 are similar for other days of the week. Constraint 8 states that the number of workers assigned consecutively paired off, plus the number of workers assigned nonconsecutively off must equal the total workforce. To avoid the assignment of two nonconsecutive days off to the same day, he introduced constraints 9 to 15. It ensures that the number of workers assigned nonconsecutively paired off on a day do not exceed the sum of remaining nonconsecutively paired off workers

Author has thus removed the possibility of two nonconsecutively off workers to be assigned to the same day as was present in Monroe [13]. But we have to depend on a Computer program package for the solution.

Next the literature on the assignment of consecutive day-off pair is discussed.

One of the early contributions to the consecutive day-off pair scheduling was by Healy [8]. He tackled the problem by an enumeration technique and used seven basic work patterns, each of which had two consecutive days off. He defined six 'Rules of Thumb' and identified fifty combinations of seven basic work patterns, which were able to give reasonably even coverage of requirements. Procedure developed can assign, day-off pairs such that the same number of workers will be available throughout the week. The use of this technique is limited to problems in which daily employee requirement during the week remains the same.

Tibrewala, Philippe and Browne (TPB) [18], have formulated assignment of two consecutive days off per week as an integer program. They have developed a three step algorithm which identifies at each iteration one RDO pair present in the optimal solution. The procedure neither requires the large mathematical structure of integer programming, nor access to a Computer to obtain solutions.

Unfortunately, the various decision steps proposed by the authors are ambiguous Moreover, there seems to be some computational errors in the sample problem illustrated in the paper. The algorithm is efficient only for problems where the daily workforce requirements are small.

Baker [2,3] has considered two closely related variations of the consecutive day-off pair assignment problem. These variations stems from the availability or non-availability of part-time workforce to avoid slack. In both the cases, the objective is to minimize the workforce size Baker has suggested a two phase technique to arrive at an optimal schedule for both types of problem. For the details of the algorithm, the reader should refer to Baker's article referred aoove. It needs to be pointed out that the problem solved by Baker using his full-time workforce algorithm [3], is numerically inconsistent.

2.2 Rotating Schedules:

Rotating schedules are used to avoid any preferential day-off pairs to be assigned always to an individual worker This is achieved by rotating workers among different day-off patterns. In certain cases, the rotation of the worker from one pattern to another can produce very long or very short consecutive workday runs for the worker. For example an employee assigned Sunday and Monday off in one week followed by Friday and Saturday in the next faces ten consecutive days at work. Upper and lower limits on such workday runs are commonly included in the labour-management agreements, and a rotating schedule must recognize such constraints.

Monroe [13] has suggested a heuristic methodology of rotating schedules for baggage handlers. His approach does not consider the upper and lower limits on workday runs for the workers. This difficulty has been overcome by Bodin [5] who proposed a network approach for finding the rotating schedules. Bodin's model is based on a planning cycle of few weeks, after which the rotating patterns of assignments repeat. His formulation involves finding a complete tour of given length in a network.

Another contribution in the area of rotating schedules have been made by Maier-Rothe and Wolfe [12], who developed an interesting heuristic approach for the generation of a number of feasible alternatives for the allocation of nursing staff.

In Chapter III, we shall present the three proposed models for day-off scheduling problems.

CHAPTER III

PROBLEM FORMULATION

In this chapter mathematical formulations of the employee day-off assignment problem is presented. Explicitly the problem involves the scheduling of workforce to meet varying cyclic requirements such that each worker gets two days off during the week. The objective is to minimize the workforce size, satisfying all manpower requirements Three approaches which satisfy different structural variations of the basic day-off scheduling problem are proposed here The approaches have been titled as, Requirements Reduction Technique (RRT), Artificially Increased Requirements Technique (AIRT), and Status Matrix Technique (SMT). The first two approaches, i.e., RRT and AIRT yield optimal schedule ensuring two consecutive days off per week for every worker. However, AIRT approach generates alternate optimal schedules for the creation of rotating schedules In the third approach, i.e. SMT two consecutive days off is considered as desirable but not essential The proposed three techniques are discussed separately in the following sections. A few illustrative problems have been added towards the end of the discussion on each technique to provide an insight into the various decision making steps.

3.1 Requirements Reduction Technique (RRT)

In this approach we determine the minimum workforce size required to satisfy the daily manpower requirements such that each worker gets two consecutive days off

The following notations are used for the mathematical formulation,

r_j	$j = 1, .. , 7$	manpower requirement for the j th day
x_j	$j = 1, ..., 7$	number of workers given off on two consecutive days, i.e , j and j+1 [illegible]
x^*_j	$j = 1, .. , 7$	optimal number of workers given off on two consecutive days. j - j+1.
X		total number of workers given off on two consecutive days
X^*		optimal workforce size
a_j	$j = 1, ..., 7$	number of workers assigned to work for the day j.
M		maximum manpower requirement during the week; $M = r_{max}$
(r_k, r_{k+1})		a day-off pair selected, which is present in the optimal solution

3.1.1 Mathematical Formulation

Since each worker has to be given off on two consecutive days, the problem of minimizing the workforce size covering the daily manpower requirements is equivalent to the problem of minimization of the total number of workers given off on two consecutive days. Let a_j be the sum of the workers off on various day-off pairs excluding the pairs covering the j-th day. Then a_j should be greater than or equal to r_j, where r_j is the workforce requirement for the j-th day. Mathematically the above stated problem can be expressed in the form of following relationships

$$\begin{array}{llcr}
\text{Minimize } X & = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 & & \\
\text{subject to,} & \phantom{x_1 +{}} x_2 + x_3 + x_4 + x_5 + x_6 & \geq & r_1 \\
 & + x_3 + x_4 + x_5 + x_6 + x_7 & \geq & r_2 \\
 & x_1 + x_4 + x_5 + x_6 + x_7 & > & r_3, \\
 & x_1 + x_2 + x_5 + x_6 + x_7 & \geq & r_4 \\
 & x_1 + x_2 + x_3 + x_6 + x_7 & \geq & r_5 \\
 & x_1 + x_2 + x_3 + x_4 + x_7 & \geq & r_6, \\
 & x_1 + x_2 + x_3 + x_4 + x_5 & \geq & r_7, \\
 & \qquad x_j \geq 0, \quad j = 1, .. 7. & &
\end{array}$$

The special mathematical structure of this problem can be utilized to advantage. The various decision steps involved in the solution of the problem are presented in the next section.

3.1.2 RRT Algorithm - Decision Steps:

At each iteration, the algorithm locates a day-off pair present in the optimal solution (Step 1). Then we artificially reduce the daily manpower requirements for all days except the days covered by the located day-off pair (Step 2). This process is continued until all requirements are reduced to zero (Step 3). Day-off pairs identified by this process indicate the optimal schedule to the given problem. The various decision steps are listed below systematically.

Step 1. (A) Locate the highest manpower requirement during the week, the next to the highest and the third largest, etc. until a unique day-off pair with unlocated manpower requirements is identified

(B) In case of a tie, choose the day-off pair with the least requirement on the adjacent day

(C) If a tie still exists, choose the first of the available pairs.

Step 2. Increase x_k^* (k chosen in Step 1) and decrease r_j by 1 for $j \neq k, k+1$ and $r_j \neq 0$.

Step 3. Repeat Steps 1 and 2 until all the requirements are reduced to zero.

Algorithm outlined above, gives an optimal schedule within finite number of iterations. In Step 1(A) we actually find out a day-off pair (k, k+1) such that

$\max.\ (r_k, r_{k+1}) = \min_j.[\max.\ (r_j, r_{j+1})] = m.$ If there exists ties, we apply Step 1 (B) and (C) to resolve them

In Step 2 we artificially reduce the manpower requirements by assigning the located day-off pair

In order to prove that the above algorithm would yield an optimal day-off schedule, we first establish that every pair selected by the use of algorithm will be present in the optimal solution (Theorems 1 and 2), and then we will show that the reduction of requirements in each iteration is justified (Theorem 3).

3.1.3 Optimality Proof of RRT Algorithm.

Theorem 1: If $m = \min_j [\max. (r_j, r_{j+1})]$ then, for any optimal solution, the objective function $X^* = \sum_{j=1}^{n} x_j^* > m$

Proof Let $M = \max. (r_j)$. If $M > m$ then $X^* \geq \max_j(r_j) = M > m$.
If $M = m$, at least one element of every pair (r_j, r_{j+1}) equals M. This means, after assigning any x_p, the new value M' will equal $\max.(r_p, r_{p+1}) = M$ Thus $X^* \geq M'+1 = M+1 = m+1$ Hence $X^* > m$.

Theorem 2: If a pair (r_k, r_{k+1}) is chosen in accordance with Step 1 of the proposed algorithm, then an optimal solution exists with $x_k \geq 1$.

Proof If a unique pair (k, k+1) is obtained in the Step 1(A) of the algorithm, $\max.(r_{k-1}, r_k) > \max. (r_k \; r_{k+1})$. Therefore $r_{k-1} > r_{k+1}$. If (B) of the Step 1 is used, then also $r_{k-1} > r_{k+1}$, as we select the pair with the least requirement on the adjacent day. In case of (C) the result is same as (A), since the preceeding pair is not minimal. Also note that $r_{k+2} \geq r_k$ when a unique (k, k+1) is chosen in Step 1 (A), (B) or (C)

Now suppose that X^* is an optimal solution with $x_k^* = 0$. Since $x_k^* = 0$, $a_k^* \geq a_{k-1}^*$ and $a_{k+1}^* \geq a_{k+2}^*$ Now,

$$a_k^* + a_{k+1}^* \geq a_{k-1}^* + a_{k+2}^* \geq r_{k-1} + r_{k+2} > r_{k+1} + r_k$$

This inequality implies that one of the following is true.

(a) $a_k^* > r_k$ and $a_{k+1}^* > r_{k+1}$,

(b) $a_k^* > r_k$ and $a_{k+1}^* = r_{k+1}$, or

(c) $a_k^* = r_k$ and $a_{k+1}^* > r_{k+1}$.

It can be easily shown for each that any one if true, a new optimal solution can be obtained by making $x_k^* = 1$ and reducing other x_j^* by 1. This is also true if all requirements are equal or alternately equal.

If $a_{k+1}^* = r_{k+1}$ and $x_{k+1}^* = 0$, then $a_{k+1}^* = r_{k+1} = X^*$ as above (Case (3)) in contradiction to Theorem 1 Hence, $x_{k+1}^* \geq 1$, andcan be reduced by 1, to obtain a new optimal solution with $x_k \geq 1$.

Theorem 3: Given a set of requirements R, if a pair (r_k, r_{k+1}) is selected in accordance with algorithm Step 1, and the requirements are reduced to R' as in Step 2, then the optimal value X^* of the original problem is $X'^* + 1$ where X'^* is the optimal solution to the reduced problem

Proof: Suppose that $X^* = X'^*$. By Theorem 2, an optimal solution exists with $x_k^* \geq 1$. Define $x_k'^* = x_k^* - 1$ and $x_j'^* = x_j^*$ for $j \neq k$. Then X'^* defined in this way must satisfy R'. The value of this solution is $X'^* - 1$, a contradiction to the original assumption

Thus the algorithm proposed leads to an optimal schedule.

3.1.4 Illustrative Examples

In order to explain the various steps involved in the proposed algorithm, two illustrative problems are presented below

Problem 1: Suppose that the weekly manpower requirements are as follows,

Monday - 8, Tuesday - 7, Wednesday - 7, Thursday - 7
Friday - 9, Saturday - 5, Sunday - 3.

Employees are to be given two consecutive days off. Develop an optimal workforce schedule satisfying all requirements

Procedure. The solution procedure involves following steps

Construct a table with daily workforce requirement values arranged in the decreasing order.

Step 1: (a) Select the highest workforce requirement value. Examine how many day-off pairs can be obtained after neglecting the selected value. If a unique day-off pair is located [it is an optimal pair according to Step 1 (Case A) of the algorithm], proceed to Step 2. If more than one day-off pairs are obtained, go to Step 1(a) neglecting the requirement values selected in previous iterations If no day-off pair is obtained go to Step 1(b).

(b) Identify the last set of requirement values selected in Step 1(a), the deletion of which resulted in no day-off pair. Identify the days corresponding to these requirement values. For each one of these days examine the requirement value of the adjacent day. Select the day/s for which the adjacent day requirement value/s is/are minimum. If a unique day-off pair is obtained go to Step 2, otherwise go to Step 1(c).

(c) Out of the tied day-off pairs, select the day-off pair which appears first during the week Go to Step 2.

Step 2: Reduce the requirement values by one from all the days excepting those covered by the selected day-off pair and those requirements which are already zero.

Step 3: Go to Step 1(a) and continue this procedure till all the requirements are reduced to zero

Table 1 gives the various iterations needed for the above stated problem. For the first four iterations the selection of a unique day-off pair is exhibited by underlining the neglected requirement values.

The optimal schedule developed is as follows:

Number of workers	Days off	Worker's Identification Number
1	Monday - Tuesday	9
2	Tuesday - Wednesday	4, 10
1	Wednesday - Thursday	6
5	Saturday - Sunday	1,2,3,5,8
1	Sunday - Monday	7

The developed schedule can be checked as to whether it satisfies all the manpower requirements during the week or not. This is done by calculating the total number of workers

Table 1: Iterative Development of RRT Algorithm (Problem 1)

S.N.	Mon.	Tue.	Wed.	Thu	Fri.	Sat.	Sun	Case	First Day of Day Off Pair k
1	$\underline{8}$	$\underline{7}$	$\underline{7}$	$\underline{7}$	$\underline{9}$	5	3	A	6
2	$\underline{7}$	$\underline{6}$	$\underline{6}$	$\underline{6}$	8	5	3	A	6
3	$\underline{6}$	$\underline{5}$	$\underline{5}$	$\underline{5}$	$\underline{7}$	5	3	B	6
4	$\underline{5}$	4	4	$\underline{4}$	$\underline{6}$	$\underline{5}$	3	C	2
5	4	4	4	3	5	4	2	B	6
6	3	3	3	2	4	4	2	B	3
7	2	2	3	2	3	3	1	B	7
8	2	1	2	1	2	2	1	C	6
9	1	0	1	0	1	2	1	C	1
10	1	0	0	0	0	1	0	C	2
11	0	0	0	0	0	0	0	0	0

$x_1^* = 1$, $x_2^* = 2$, $x_3^* = 1$, $x_4^* = 0$, $x_5^* = 0$, $x_6^* = 5$, $x_7^* = 1$, $X^* = 10$.

Table 2. Daily Schedule

x_j^*	Mon.	Tue.	Wed.	Thu	Fri	Sat	Sun
1	0	0	1	1	1	1	1
2	2	0	0	2	2	2	2
1	1	1	0	0	1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
5	5	5	5	5	5	0	0
1	0	1	1	1	1	1	0
a_j	8	7	7	9	10	5	4
r_j	8	7	7	7	9	5	3

assigned for each day. Table 2 gives the daily schedule for the workers.

From Table 2 it is clear that the daily manpower requirements for all the days of the week are satisfied

Problem Statement 2:

Employees enjoy two consecutive days off per week. Schedule minimum workforce satisfying following demands,

Monday - 17, Tuesday - 13, Wednesday - 15, Thursday 19,
Friday - 14, Saturday - 16, Sunday - 11

Solution Procedure.

Procedure and developed optimal schedule is exhibited in Table 3.

Table 3: Iterative Development of RRT Algorithm (Problem 2)

S.N.	Mon.	Tue.	Wed	Thu	Fri.	Sat.	Sun.	Case	First day of Day-Off pair, r
1	17	13	15	19	14	16	11	A	2
2	16	13	15	18	13	15	10	B	6
3	15	12	14	17	12	15	10	A	2
4	14	12	14	16	11	14	9	C	6
5	13	11	13	15	10	14	9	B	7
6	13	10	12	14	9	13	9	A	2
7	12	10	12	13	8	12	8	C	5
8	11	9	11	12	8	12	7	B	7
9	11	8	10	11	7	11	7	A	2
10	10	8	10	10	6	10	6	C	4
11	9	7	9	10	6	9	5	C	6
12	8	6	8	9	5	9	5	B	7
13	8	5	7	8	4	8	5	A	2
14	7	5	7	7	3	7	4	C	4
15	6	4	6	7	3	6	3	C	5
16	5	3	5	6	3	6	2	B	7
17	5	2	4	5	2	5	2	A	2
18	4	2	4	4	1	4	1	C	4
19	3	1	3	4	1	3	0	C	6
20	2	0	2	3	0	3	0	C	7
21	2	0	1	2	0	2	0	A	2
22	1	0	1	1	0	1	0	C	4
23	0	0	0	1	0	0	0	C	2

$x_1^* = 0$, $x_2^* = 7$, $x_3^* = 0$, $x_4^* = 4$, $x_5^* = 3$, $x_6^* = 4$, $x_7^* = 5$, $X^* = 23$.

Table 3: Iterative Development of RRT Algorithm (Problem 2)

S.N.	Mon.	Tue	Wed	Thu	Fri.	Sat	Sun	Case	First day of Lo-Off pair, i
1	17	13	15	19	14	16	11	A	2
2	16	13	15	18	13	15	10	B	6
3	15	12	14	17	12	15	10	A	2
4	14	12	14	16	11	14	9	C	6
5	13	11	13	15	10	14	9	B	7
6	13	10	12	14	9	13	9	A	2
7	12	10	12	13	8	12	8	C	5
8	11	9	11	12	8	12	7	B	7
9	11	8	10	11	7	11	7	A	2
10	10	8	10	10	6	10	6	C	4
11	9	7	9	10	6	9	5	C	6
12	8	6	8	9	5	9	5	B	7
13	8	5	7	8	4	8	5	A	2
14	7	5	7	7	3	7	4	C	4
15	6	4	6	7	3	6	3	C	5
16	5	3	5	6	3	6	2	B	7
17	5	2	4	5	2	5	2	A	2
18	4	2	4	4	1	4	1	C	4
19	3	1	3	4	1	3	0	C	6
20	2	0	2	3	0	3	0	C	7
21	2	0	1	2	0	2	0	A	2
22	1	0	1	1	0	1	0	C	4
23	0	0	0	1	0	0	0	C	2

$x_1^* = 0$, $x_2^* = 7$, $x_3^* = 0$, $x_4^* = 4$, $x_5^* = 3$, $x_6^* = 4$, $x_7^* = 5$, $X^* = 23$.

3.2 Artificially Increased Requirements Technique (AIRT).

This approach has been developed for the generation of multiple optimal schedules for the two consecutive day-off scheduling problem These alternate optimal schedules can be used for providing rotating schedules for the workforce If the management desires to implement only one fixed schedule it can be selected out of the generated optimal schedules based on some practical considerations.

The proposed algorithm takes advantage of the inherent slack present in any workforce allocation to the problem. The slack gets introduced into the solution of the consecutive day-off scheduling problem if the following characteristics are present,

1 Sum of the daily manpower requirements is not a multiple of five, and/or

2. maximum manpower requirement on a particular day of the week is greater than one fifth of the total daily manpower requirements.

The following notations have been used for the problem formulation,

r_j $j = 1, \ldots, 7$ manpower requirement for the j-th day of the week

b_j $j = 1, \ldots, 7$ maximum number of workers who could be assigned the j-th day off.

3.2 Artificially Increased Requirements Technique (AIRT)

This approach has been developed for the generation of multiple optimal schedules for the two consecutive day-off scheduling problem. These alternate optimal schedules can be used for providing rotating schedules for the workforce. If the management desires to implement only one fixed schedule, it can be selected out of the generated optimal schedules based on some practical considerations.

The proposed algorithm takes advantage of the inherent slack present in any workforce allocation to the problem. The slack gets introduced into the solution of the consecutive day-off scheduling problem if the following characteristics are present,

1 Sum of the daily manpower requirements is not a multiple of five, and/or

2. maximum manpower requirement on a particular day of the week is greater than one fifth of the total daily manpower requirements.

The following notations have been used for the problem formulation,

r_j $j = 1, \ldots, 7$ manpower requirement for the j-th day of the week

b_j $j = 1, \ldots, 7$ maximum number of workers who could be assigned the j-th day off.

x_j $j = 1, \ldots, 7$ number of workers given off on two consecutive days, i.e , j and (j + 1)days

W minimal workforce size

t amount of slack

u increase in the workforce size to bring feasibility into the solution

S_j $j = 1, \quad , 7$ set of three nonadjacent indices among the integers 1,. ., 7 excluding j and j+1 (7+1 treated as 1). viz. $S_1 = [3,5,7]$, $S_5 = [2,4,7]$, etc

y_j $j = 1, \quad , 7$ change in day-off allowance.

$\langle a \rangle$ smallest integer greater than or equal to a.

Parameters of the modified problem, used to bring feasibility in the solution, are identified by primes

The mathematical formulation of the day-off scheduling problem, which can be solved by AIRT algorithm, is given in the next section.

3.2.1 <u>Mathematical Formulation.</u>

In any no-slack schedule, the sum of the workers assigned to two consecutive day-off pairs (viz. j-1 and j), must be equal to the maximum number of workers who could be assigned the day j off. Thus,

$$x_7 + x_1 = b_1, \qquad (1)$$
$$x_{j-1} + x_j = b_j, \; 2 \leq j \leq 7$$

where b_j is the day-off allowance

$$b_j = w-r_j \qquad (2)$$

The solution to the set of equations (1) is given by,

$$x_j = \tfrac{1}{2} [\sum_{i \notin S_j} b_i - \sum_{i \in S_j} b_i],$$

viz. $$x_j = \tfrac{1}{2} \sum_{j=1}^{7} b_j - \sum_{i \in S_j} b_i \qquad (3)$$

Above relationship is important as it gives us the value of x_j.

Employees enjoy five-day workweek. Hence, if $\sum_{j=1}^{7} r_j$ is not a multiple of 5, there would be some slack present in any allocation of workers. No solution could improve on a workforce size of $\frac{1}{5} \sum_{j=1}^{7} r_j$, rounded to the next higher integer. Further, the workforce size should at least be equal to the peak manpower requirement during the week.

Hence the minimum workforce size is given by,

$$W = \max. \left[\left\langle \frac{1}{5} \sum_{j=1}^{7} r_j \right\rangle , r_{\max} \right]. \qquad (4)$$

Let t be the amount of slack corresponding to the minimal workforce, then

$$t = 5W - \sum_{j=1}^{7} r_j \qquad (5)$$

Original problem should be modified so that slack appears in it, in the form of artificially increased requirements The optimal solution to the original problem will correspond to a no-slack solution to the modified problem, in which staffing requirements are artificially increased and day-off allowances altered by an amount y_j.

Thus, $b_j' = b_j + y_j$

Let u denote the increase in workforce size to bring feasibility in the original problem,

$$W' = W + u .$$

Treating y_j as decision variables the objective is to minimize the increase in the workforce size such that $x_j' \geq 0$ for the modified problem

$$\begin{aligned} \text{Since,} \quad W + u - r_j' &= W' - r_j' , \\ &= b_j' , \\ &= b_j + y_j , \\ \text{viz.} \quad W + u - r_j &= W - r_j + y_j , \end{aligned}$$

The various decision steps involved in the algorithm are presented in the next section.

3.2.2 AIRT Algorithm:

1. Calculate the value of minimal workforce size.

$$W = \max.\ [\langle \tfrac{1}{5} \sum_{j=1}^{7} r_j \rangle ,\ r_{\max.}\ ] . \qquad (4)$$

2. Calculate the value of slack:

$$t = 5W - \sum_{j=1}^{7} r_j \qquad (5)$$

3. Calculate b_j i.e. day-off allowances and x_j i.e. workers scheduled:

$$b_j = W - r_j, \qquad (2)$$

$$x_j = \tfrac{1}{2} \sum_{j=1}^{7} b_j - \sum_{i \varepsilon S_j} b_i . \qquad (3)$$

4. Calculate the value of u i.e. increase in workforce size:

$$u \geqq \max.\ [0, \langle (-x_{\min.} -t/2)/3 \rangle]. \qquad (11)$$

5. The following linear program is solved for the values of y_j with the help of a tableau format:

$$\text{minimize } u = \text{maximize } [y_j], \qquad (6)$$

$$b_j' = b_j + y_j \geqq 0 \qquad (10)$$

The above stated problem has linear constraints and objective function. Thus it can be formulated as a Linear Programming Problem.

Mathematically the problem can be expressed in the following set of relationships,

$$\text{minimize } u = \text{maximize } [y_j], \qquad (6)$$

subject to,
$$\sum_{j=1}^{7} y_j = 2u - t,$$

$$\sum_{i \varepsilon S_j} y_i \leqq x_j + u - \frac{t}{2}, \quad 1 \leqq j \leqq 7, \qquad (8)$$

$$y_j \geqq -b_j, \qquad 1 \leqq j \leqq 7, \qquad (9)$$

where y_j are integer variables. (10)

By the algebraic manipulation of the first two cons traints (i.e equations (8) and (9)), we get the generalized bound,

$$u \geqq \text{max}, [\ 0, \langle(-x_{\text{min}}. - t/2) / 7\rangle]. \qquad [illegible]$$

Any schedule satisfying the above mentioned constraints is a basic feasible solution to the modified problem and is optimal. By searching the different sets of y_j values, alternate optima can be obtained. Thus many optimal schedules can be generated and used as rotating schedules.

[illegible] involved in the algorithm are [illegible].

3.2.2 ALGORITHM

1 Calculate [illegible] workforce size

$$[illegible] \; \frac{1}{[illegible]} \sum [illegible] \rangle , \; r_{max} \;] \qquad (4)$$

2 Calculate [illegible]

$$[illegible] \qquad (5)$$

3. Calculate [illegible] allowances and y_j i.e. workers [illegible]

$$[illegible] \qquad (2)$$

$$[illegible] \qquad (3)$$

4 Calculate [illegible] increase in workforce size

$$[illegible] = [illegible]x.\,[0, \langle([illegible]_{min.} -t/2)/3\rangle] \qquad (11)$$

The following linear program is solved for the values of y_j with the help of a tableau format.

$$\text{minimize } [illegible] = \text{maximize } [y_j], \qquad (6)$$

Initial schedule developed is not feasible as some of the x_j's are negative. Hence, constraints are expressed in a tableau format, so as to determine y_j values. Tableau format is exhibited in Table 5.

Table 5: Solution of Linear Programming Problem.

Row B	-15	-5	-20	0	-20	-5	-15	
Row Y	- 5	0	-15	0	-15	0	- 5	-40
Row 1	▨	▨	-15	▨	-15	▨	- 5	-35
Row 2	- 5	▨	▨	0	▨	0	▨	0
Row 3	▨	0	▨	▨	-15	▨	- 5	-20
Row 4	- 5	▨	-15	▨	▨	0	▨	-20
Row 5	▨	0	▨	0	▨	▨	- 5	0
Row 6	- 5	▨	-15	▨	-15	▨	▨	-35
Row 7	▨	0	▨	0	▨	0	▨	10

In this tableau Row B lists $-b_j$ values, and Row Y indicates y_j values; along with 2u-t value in the 8th column. Next is, a set of 7 constraints with some cells hatched and some blank. First seven columns of this set with blank cells indicate elements belonging to S_j set and hatched indicate the elements not belonging to S_j set.

Solution Procedure:

An examination of the problem reveals that although the sum of the daily manpower requirements is a multiple of 5, we will have to introduce slack due to the peak manpower requirement of 20 on Thursday. Hence,

$$W = \max. \left[\left\langle \frac{1}{5} \sum_{j=1}^{7} r_j \right\rangle, r_{\max} \right], \qquad (4)$$

$$= 20,$$

$$\text{slack}, \quad t = 5W - \sum_{j=1}^{7} r_j, \qquad (5)$$

$$= 40.$$

Day off allowances can be obtained by subtracting the requirement values from the workforce size. Also number of workers who could be assigned day-off pairs, can be determined from the set of recursive equations (1). Calculations are exhibited in Table 4.

Table 4: Calculation of Initial Schedule

W	Day j	r_j	b_j	x_j
20	1	5	15	-15
	2	15	5	20
	3	0	20	0
	4	20	0	0
	5	0	20	20
	6	15	5	-15
	7	5	15	30

$$\sum_{j=1}^{7} r_j = 60, \quad t = 40, \quad u = 0$$

Initial schedule developed is not feasible as some of the x_j's are negative. Hence, constraints are expresseo in a tableau format, so as to determine y_j values Tableau format is exhibited in Table 5

Table 5 Solution of Linear Programming Problem

Row B	-15	-5	-20	0	-20	-5	-15	
Row Y	- 5	0	-15	0	-15	0	- 5	-40
Row 1	▨	▨	-15	▨	-15	▨	5	-35
Row 2	- 5	▨	▨	0	▨	0	▨	0
Row 3	▨	0	▨	▨	-15	▨	- 5	- 20
Row 4	- 5	▨	-15	▨	▨	0	▨	-20
Row 5	▨	0	▨	0	▨	▨	- 5	0
Row 6	- 5	▨	-15	▨	-15	▨	▨	-35
Row 7	▨	0	▨	0	▨	0	▨	10

In this tableau Row B lists $-b_j$ values and Row Y indicates y_j values, along with 2u-t value in the 8th column. Next is, a set of 7 constraints with some cells hatched and some blank. First seven columns of this set with blank cells indicate elements belonging to S_j set and hatched indicate the elements not belonging to S_j set.

With the help of r_j values found out from Table 5 we can proceed for the solution of the modified problem. Calculations are exhibited in Table 6, along with the optimal schedule. This is a no-slack solution to the modified problem with the artificially increased requirements

Table 6: Calculation of Modified Schedule

W'	Day j	r'_j	b'_j	x'
20	1	10	10	0
	2	15	5	5
	3	15	5	0
	4	20	0	0
	5	15	5	5
	6	15	5	0
	7	10	10	10

Many more schedules can be developed by generating various sets of y_j values satisfying all constraints. Few of the generated schedules are given in Table 7

Table 7. Alternate Optimal Schedules

x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	5	0	0	0	0	15
0	5	0	0	4	1	10
1	4	0	0	0	5	10
1	4	0	0	1	0	14
0	4	0	0	1	4	11
0	0	0	0	5	0	15
0	5	0	0	0	5	10
1	4	0	0	0	1	14
2	3	0	0	2	3	10
5	0	0	0	5	0	10

These schedules if used in succession, develop a rotating schedule.

3 3 Status Matrix Technique (SMT)

The main characteristics of the Status Matrix Technique, are as follows.

1. The workforce size is given and is approximately equal to the demand for the staff Normally each worker will work for 5 days in a week. However, depending upon workload, the workers may be assigned 4 or 6 working days per week

2 Consecutive off days are desirable, but not essential As far as possible non-consecutive day-off patterns are to be avoided. If it is not practicable, allow maximum workdays in between two off days

3 Weekend off days are to be assigned on an equitable, rotating basis.

4. Model can incorporate employee preferences in the schedule to be developed.

Status Matrix Technique explained here is flexible enough to take into consideration various aspects in day-off scheduling which creep up in real life situations, viz preassigned allocations, leave, overtime, etc. Here it is aimed to allocate only the required number of workers, assuming flexibility in the day-off assignment.

The notations used in the problem formulation are given below.

W		workforce size
da		off day assigned
q		requested off day assigned
O_i	$i=1, \quad , W$	off days assigned to the i th employee in a week, (Status Variable)
O_j	$j-1, . \quad , 7$	off days available (to be assigned on the j-th day, (Status Variable
S_{ij}	$i=1, . \quad , W$ $j=1, \ldots, 7$	status of the i-th employee on the j-th day, (Status Matrix).

3.3.1 Mathematical Formulation.

As the number of off days per week is two, the sum of off days available minus two times the workforce size gives expected surplus or overtime days

$$\sum_{j=1}^{7} O_j - 2W = \begin{cases} + & \text{surplus days, (a third off day)} \\ 0 & \text{even workforce, (all receive 2 off} \\ - & \text{overtime days, (one off day only)} \end{cases}$$

Also off days assigned to an individual employee must be normally 2 in a week. Exceptions may arise due to overtime or surplus.

$$\sum_{j=1}^{7} da_i = \begin{cases} 1 & \text{overtime,} \\ 2 & \text{normally,} \\ 3 & \text{surplus.} \end{cases}$$

Number of workers assigned off on a particular day must be equal to the number of off days available

$$\sum_{i=1}^{W} da_{j} = O_{j}$$

It is desired to give weekends off on a rotational basis. In a rotation list employees numbers are arranged in some specific order such as alphabetical, numerical on employee numbers or otherwise randomized sequence. Pre-assigned allocations, employee preferences for off days, etc are to be accommodated in the final schedule. Interval between two nonconsecutive off days is to be maximized Status variables and status matrix indicate day-off assignment

Various steps in the algorithm developed are listed below.

3.5.2 Status Matrix Technique (SMT) Algorithm

1 Calculate the off days available for each day.

2. Calculate the surplus or overtime days

3. Initialize the status matrix and variables assuming that all employees are working on all days

4. Accept and record all over-riding information like training, urgent need, etc

5. Assign weekend off days according to rotation list

6 Assign requested off days to the employees, based on preference rank and order in which request is received

7 Assure that all employees are assigned at least one off day

8 Achieve maximum consecutive off days, by checking the available off days before and after the assigned off day

9. Assign the remaining off days with maximum workdays in between

10 Mutual exchange of off days is permitted and carried out in the end

Procedure outlined above always keeps track of the available off days for assignment on a particular day and individual employee day-off status

An illustrative problem is presented in the next sectio

3.3.3 Illustrative Example

Problem Statement.

Schedule a workforce of 16 employees to satisfy the daily manpower requirements as indicated below.

Sunday - 5, Monday - 15, Tuesday - 13, Wednesday - 12
Thursday - 12, Friday - 14, and Saturday - 8

Assume that first day of the operating week is Sunday. Each employee is to be given two off days in a week preferabl on consecutive days. If nonconsecutive off days are to be assigned allow maximum workdays in between. Weekend days off are to be given on equitable basis to all the employees. Each employee has an assigned identification number (1 to 16). The off days assignment to the employees is to be done according to a rotation list. The rotation list comprises of employee numbers arranged in ascending order. Employees 15 and 16 are manager and supervisor, respectively. They must be given weekends off. Assume that in the previous week the employee 2 was given Saturday off and employee 11 was given Sunday off. The preferences of the employees for weekdays off are recorded as ranks. First rank implies first preference and so on. The preferences of the employees for weekdays off are listed in Table 8.

Table 8 Employee Preferences for Weekdays Off

Order in which requests are received	Monday		Tuesday		Wednesday		Thursday		Friday	
	E	R	E	R	E	R	E	R	E	R
1	5	1					8	1	7	3
2			4	2	7	2				

E - Employee number,

R - Rank.

Solution Procedure

Let us first calculate the off days available for each day. The off days available, O_j are obtained by substracting the daily manpower requirements from the total available workforce. The number of off days available on various days are.

Sunday - 11, Monday - 1, Tuesday - 3, Wednesday - 4,
Thursday - 4, Friday - 2, Saturday - 8

Therefore, the total number of off days available during the week are 33. A workforce size of 16 employees will require 32 off days during the week Hence we have a surplus of 1 day.

A status matrix is constructed for indicating off days assigned. The off days assigned to an employee and off days available on a particular day are denoted by status variables O_i and O_j, respectively There are no entries in the status matrix initially indicating that all the employees are working on all days Table 12 indicates the final employee allocations. The procedure for making entries in the status matrix is explained below. Off days allocated are denoted by 'da'.

The assignments are made considering the over-riding status information first. Since employees 15 and 16 must be given off on Saturday and Sunday, they are assigned off days as shown in Table 12. The revised off days available for Saturday and Sunday are 6 and 9 respectively. The current values of the off days assigned to both the employees is 2.

Next, weekend days off are assigned according to the rotation list. Since employee 2 got Saturday off during the last week, the assignment of Saturday off is started with employee 3. Similarly the assignment of Sunday off is started with employee 12. The assignments are made till off days available are reduced to zero. Following this procedure we observe that employees 12 to 6 are assigned Sunday off and employees 3 to 8 are assigned Saturday off. Employees 15 and 16 are skipped as they have been already assigned two off days.

The next step is to assign requested weekdays off to the employees. The weight assigned to an employee is determined by multiplying the position and rank values of the employee. The requested weekdays off are assigned according to the ascending order of weights. The requested off days are not granted if 1) off days available on the day is zero, or 2) off days already assigned to the employee under consideration is two, or 3) more preferred requested off day can be assigned. By adopting this procedure we assign Wednesday and Thursday to employees 7 and 8, as outlined in Table 9. These assignments are denoted by 'qda' in Table 12

Table 9 Assignment Decisions Based on Requested Off Days

Request Number	Position	Rank	Weight = Position x Rank	Employee	Day	Decision
1	1	1	1	5	Monday	Refused,0
2	1	1	1	8	Thursday	Assigned,
3	1	3	3	7	Friday	Refused, rank ass
4	2	2	4	4	Tuesday	Refused,0
5	2	2	4	7	Wednesday	Assigned

At this stage we make off day assignments to employees who haven't got any. Monday, Tuesday and Wednesday are assigned as off days to employees 9, 10 and 11 respectively These assignments are also indicated in Table 12.

The procedure for maximizing consecutive off days is now undertaken. The process is initiated by selecting an employee at random (say employee 10). Off days available before and after the assigned day, are checked and if availabl one of the off days is assigned. By doing so, we are able to locate three adjacent off day assignments for employees 9, 10 and 11. The procedure carried out is shown in Table 10

Next the allocation of remaining off days, with maximum workdays in between is now undertaken. An employee is selected at random from the rotation list (say employee 9 has been selected). This assignment procedure is shown in Table 11.

Table 10 Consecutive Off Day Assignment.

Employee	Off Days Assigned	Location of first off-day	Off Days Available on Adjacent Days		Decision abo Consecutive Off Day
			Before	After	
*10	1	Tuesday	Monday 1	Wednesday 2	Assign Wedne
*11	1	Wednesday	Tuesday 2	Thursday 2	Assign Thurs
12	1	Sunday	None	Monday 1	Not availabl
13	1	Sunday	None	Monday 0	Not availabl
14	1	Sunday	None	Monday 0	Not availabl
15	2				Assigned alr
16	2				Assigned alr
1	1	Sunday	None	Monday 0	Not availabl
2	1	Sunday	None	Monday 0	Not availabl
3	2				Assigned alr
4	2				Assigned alr
5	2				Assigned alr
6	2				Assigned alr
7	2				Assigned alr
8	2				Assigned alr
*9	1	Thursday	Wednesday 1	Friday 0	Assign Wedne

CHAPTER IV

CONCLUSIONS

The work presented in this thesis deals with the scheduling of five-day workweek employees to seven-day operating week. Two days off are to be given per week to every employee. They can either be given consecutively or nonconsecutively. Three different approaches have been proposed in this work. Methodologies RRT and AIRT deals with consecutive day-off assignments where as SMT allows nonconsecutive day-off assignments. The RRT algorithm arrives at an optimal schedule satisfying daily manpower requirements and assigning two consecutive days off. The AIRT algorithm, generates alternate optimal schedules which are used for developing rotating schedules. This becomes possible due to the inherent slack present in any workforce allocation satisfying all constraints of the problem. The SMT algorithm ensures minimum workforce allocation by utilizing nonconsecutive day-off assignments. It also incorporates employee rankings, preferences etc.

The methodologies, RRT and AIRT, have been tested on the problems available in the literature. It has been found

Table 12 Final Status Matrix

Employe	Sun.	Mon	Tue	Wed.	Thu	Fri	Sat	O_i
1	da				da			2
2	da		da					2
3	da						da	2
4	da						da	2
5	da						da	2
6	da						da	2
7				qda			da	2
8					qda		da	2
9				da	da			2
10			da	da				2
11				da	da			2
12	da	da						2
13	da					da		2
14	da					da		2
15	da						da	2
16	da						da	2
O_J	0	0	1	0	0	0	0	
Off Days Assigned	11	1	2	4	4	2	8	
Employees Working	5	15	14	12	12	14	8	

CHAPTER IV

CONCLUSIONS

The work presented in this thesis deals with the scheduling of five-day workweek employees to seven-day operating week. Two days off are to be given per week to every employee. They can either be given consecutively or nonconsecutively. Three different approaches have been proposed in this work. Methodologies RRT and AIRT deals with consecutive day off assignments where as SMT allows nonconsecutive day-off assignments. The RRT algorithm arrives at a optimal schedule satisfying daily manpower requirements and assigning two consecutive days off. The AIRT algorithm, generates alternate optimal schedules which are used for developing rotating schedules. This becomes possible due to the inherent slack present in any workforce allocation satisfying all constraints of the problem. The SMT algorithm ensures minimum workforce allocation by utilizing nonconsecutive day-off assignments. It also incorporates employee rankings, preferences etc.

The methodologies, RRT and AIRT, have been tested on the problems available in the literature. It has been fou that AIRT requires lesser computational effort than Tibrewala Philippe and Browne's algorithm. The main feature of AIRT is its capability to produce rotating schedules without resorting to computer facility This is because the proposed algorithm is simpler as compared to algorithms available in the literature. Moreover, AIRT yields many optimal schedules. There is no algorithm available in the literature which permits the generation of nonconsecutive day-off schedules considering the special features like employee preferences, rankings, overtime, etc Therefore the computational effectiveness of SMT could not be compared. The proposed algorithms are very simple computationally and involve only hand calculations

Workforce scheduling represent an area of scheduling with close coordination between theory and practice The existing methodologies (including the 3 methodologies proposed in this work), rely on deterministic information, even though the problems have probabilistic characteristics. Therefore, a natural extension of the present work would be to incorporate probabilistic aspects of workforce scheduling problems Further none of the approaches developed so far considers the cost aspects of workforce allocation. Hence, the proposed models need to be modified to include the cost aspects of workforce scheduling.

To conclude one may say that, a major stimulus for research efforts in manpower scheduling will continue to be the on going examination and discussion of manpower scheduling problems, as and when they arise in practice

BIBLIOGRAPHY

1. Arabeyre, J., Fearnly, J., Steiger, F. and Teather, W 'The Airline Crew Scheduling Problem. A Survey', *Transportation Science*, Vol 3, No 2 (1969)

2 Baker, K.R., 'Scheduling Full-Time and Part-Time Staff to Meet Cyclic Staffing Requirements', *Operational Research Quarterly*, Vol 25, No 1 (1974)

3. Baker, K.R., 'Scheduling a Full-Time Workforce to Meet Cyclic Staffing Requirements', *Management Science*, Vol. 20, No. 12 (August 1974)

4 Bennett, B.T and Potts, R B., 'Rotating Roster for a Transit System', *Transportation Science*, Vol 2, No 1, (1969).

5 Bodin, L., 'Towards a General Model for Manpower Scheduling - Part I and II', *Journal of Urban Analysis*, Vol 1, No. 2 (1973)

6. Brownell, W.S., and Lowerre, J.W., 'Determination and Scheduling of Workforce Required in Continuous (7 Operation', *Management Science* (to appear)

7. Butterworth, R.W. and Howard, G.T., 'A Method of Determining Highway Patrol Manning Schedules', 44th National ORSA Meeting (November, 1973)

8. Healy, W.C., 'Shift Scheduling Made Easy', *Factory* Vol. 113 No 10 (October 1959)

9. Ignall, E., Kolesar, P., and Walker, W, 'Linear Programming Models of Crew Assignment for Refuse Collections', *IEEE Transcations on SMC*, Vol. 2, No. 5 (November 1972)

10. Kolesar, P.J., Rider, K.L, Crabill, T.B, and Walker, W 'A Queueing - Linear Programming Approach to Scheduling Police Patrol Cars', Report No. WN-8761-NYC, The New York City Rand Institute (August 1974).

11 Luce, B.J., 'A Shift Scheduling Algorithm', 44th National ORSA Meeting (November 1973)

12 Maier-Rothe, C , and Wolfe, H., 'Cyclic Scheduling and Allocation of Nursing Staff', *Socio-Economic Planning Sciences*, Vol 7, No 5 (October 1973)

13 Monroe, G., 'Scheduling Manpower for Service Operations', *Industrial Engineering*, Vol. 2, No. 8 (August 1970)

14. Rothstein, M , 'Scheduling Manpower with Mathematical Programming', *Industrial Engineering*, Vol 4, No 4 (April 1972)

15 Sahney, V., 'Modelling Labour Assignment Decisions', Proceedings of 5th Annual Southeastern Symposium on System Theory (March 1973).

16. Segal, M , 'The Operator Scheduling Problem - A Network Flow Approach', *Operations Research*, Vol 22, No 4 (July-August 1974)

17. Stainer, G., *Manpower Planning*, William Heinemann Ltd London (1971).

18 Tibrewala, R., Philippe,D. and Browne, J , 'Optimal Scheduling of Two Idle Periods', *Management Science*, Vol. 19, No. 1 (September, 1972)

19. Wagner, H.M., *Principles of Operations Research*, Prentice-Hall (1969).